A New Algorithm for computing multivariate Gauss-like quadrature points

Mark A. Taylor
mataylo@sandia.gov
Sandia National Laboratory

Beth A. Wingate
wingate@lanl.gov
Los Alamos National Laboratory

Len P. Bos
lpbos@math.ucalgary.ca
Dept. of Mathematics and Statistics
University of Calgary

ABSTRACT:

The diagonal-mass-matrix spectral element method has proven very successful in geophysical applications dominated by wave propagation. For these problems, the ability to run fully explicit time stepping schemes at relatively high order makes the method more competitive then finite element methods which require the inversion of a mass matrix. The method relies on Gauss-Lobatto points to be successful, since the grid points used are required to produce well conditioned polynomial interpolants, and be high quality "Gauss-like" quadrature points that exactly integrate a space of polynomials of higher dimension than the number of quadrature points.

These two requirements have traditionally limited the diagonal-mass-matrix spectral element method to use square or quadrilateral elements, where tensor products of Gauss-Lobatto points can be used. In non-tensor product domains such as the triangle, both optimal interpolation points and Gauss-like quadrature points are difficult to construct and there are few analytic results. To extend the diagonal-mass-matrix spectral element method to (for example) triangular elements, one must find appropriate points numerically. One successful approach has been to perform numerical searches for high quality interpolation points, as measured by the Lebesgue constant (Such as minimum energy electrostatic points and Fekete points). However, these points typically do not have any Gauss-like quadrature properties.

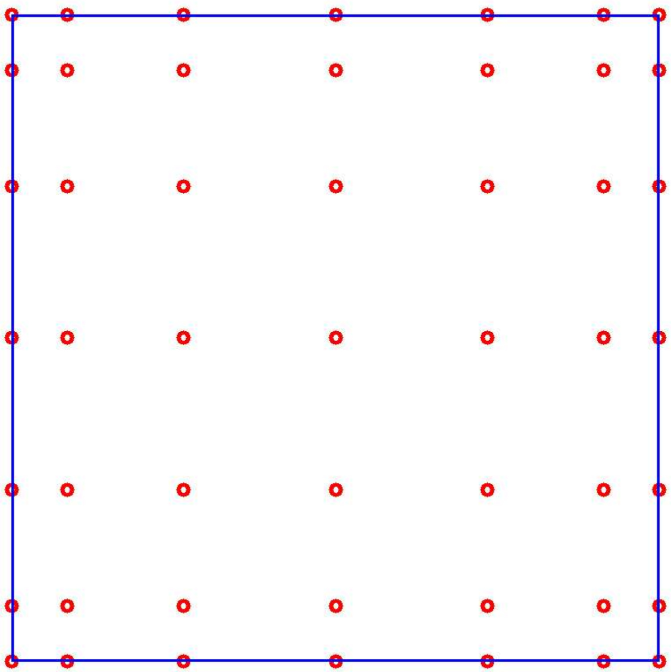
In this work, we describe a new numerical method to look for Gauss-like quadrature points in the triangle, based on a previous algorithm for computing Fekete points. Performing a brute force search for such points is extremely difficult. A common strategy to increase the numerical efficiency of these searches is to reduce the number of unknowns by imposing symmetry conditions on the quadrature points. Motivated by spectral element methods, we propose a different way to reduce the number of unknowns: We look for quadrature formula that have the same number of points as the number of basis functions used in the spectral element method's transform algorithm. This is an important requirement if they are to be used in a diagonal-mass-matrix spectral element method.

This restriction allows for the construction of cardinal functions (Lagrange interpolating polynomials). The ability to construct cardinal functions leads to a remarkable expression relating the variation in the quadrature weights to the variation in the quadrature points. This relation in turn leads to an analytical expression for the gradient of the quadrature error with respect to the quadrature points. Thus the quadrature weights have been completely removed from the optimization problem, and we can implement an exact steepest descent algorithm for driving the quadrature error to zero.

Results from the algorithm will be presented for the triangle and the sphere.

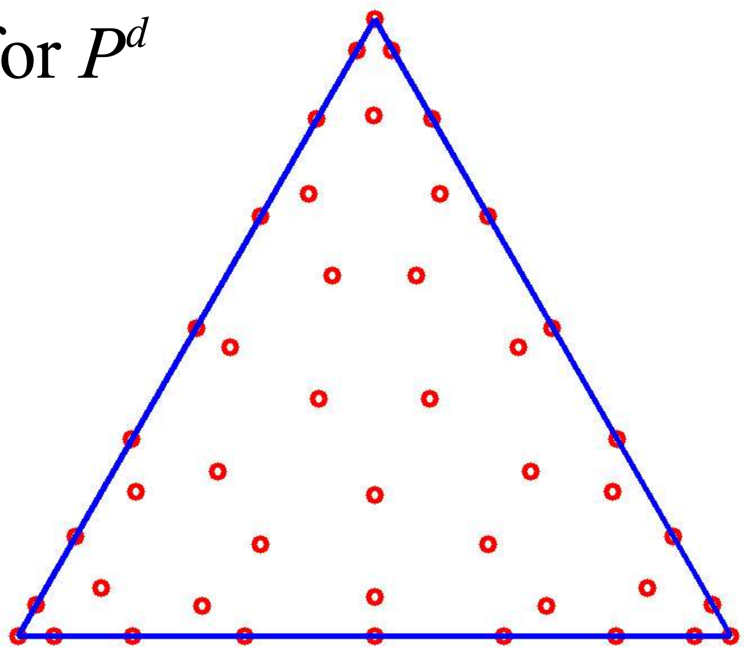
Diagonal-mass-matrix spectral element methods

- Nodal sets for quadrilaterals:
  - $P^d = \text{span}\ \{\ x^m y^n : n<d,\ m<d\ \}$
  - Tensor product of Gauss-Lobatto points
  - Cardinal function basis $\{\phi_i\}$ for P^d
  - Small Lebesgue constant
  - Gauss-like quadrature



Diagonal-mass-matrix spectral element methods

- Nodal sets for triangles:
  - $P^d = \text{span}\ \{\ x^m y^n : n+m<d\}$
  - Points are not known analytically
  - Cardinal function basis $\{\phi_i\}$ for P^d
  - Small Lebesgue constant?
  - Gauss-like quadrature?



Lebesgue Constant $\|I\|_\infty$

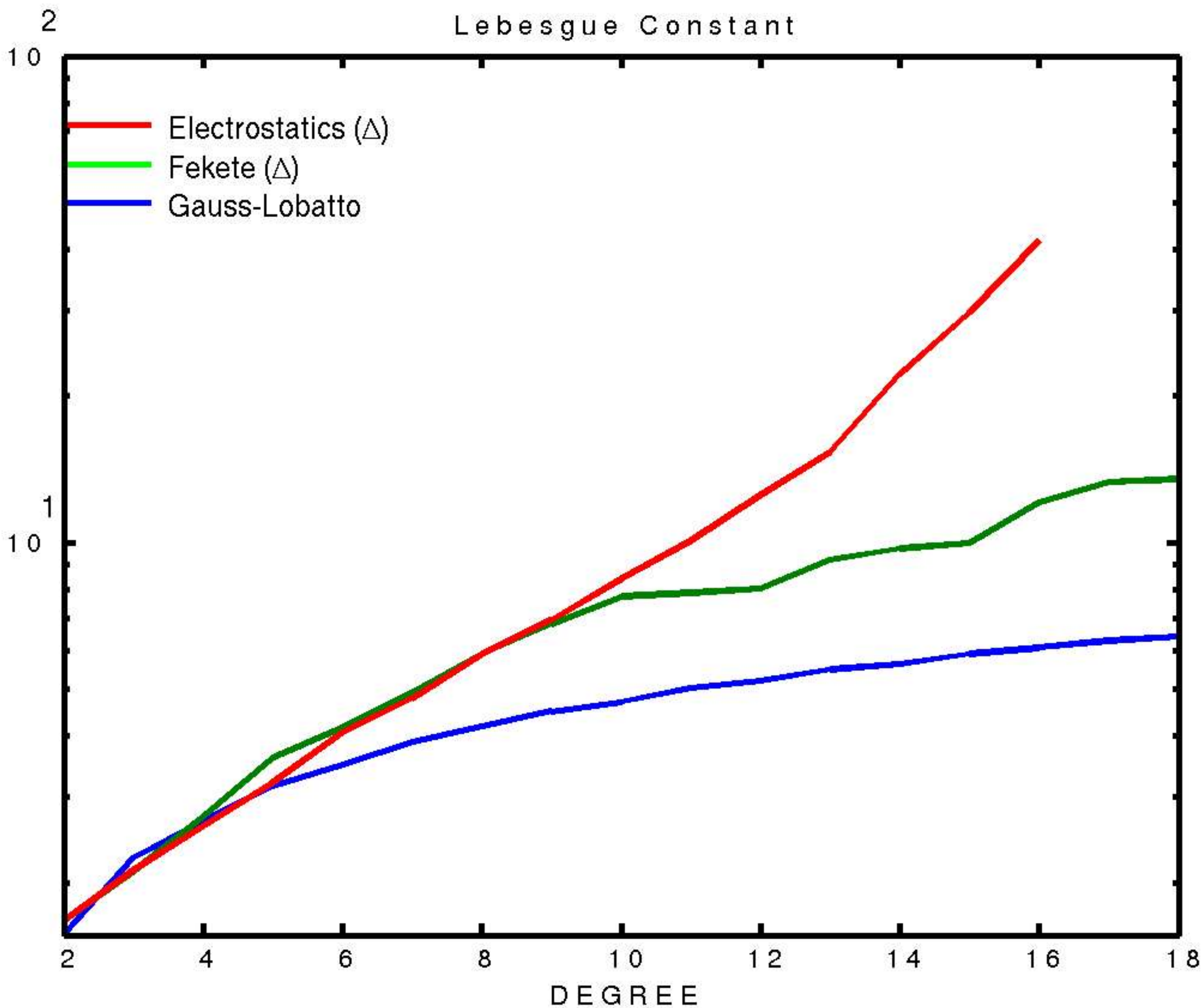
Max norm of interpolation operator I:

$$\|f - I(f)\| \leqslant \|I\| \; \|f - g\| \qquad \forall\, g \in P^d$$

Small Lebesgue constant = well conditioned cardinal function basis:

$$\|I\| = \max_{z \in \Delta} \sum_i |\phi_i(z)|$$

Lebesgue Constant



Hesthaven, *From electrostatics to Almost Optimal Nodal Sets for Polynomial Interpolation In A Simplex*, SIAM J. Numer Anal, 1998
Taylor, Wingate, Vincent, *An algorithm for computing Fekete points in the triangle*, SIAM J. Numer. Anal., 2000

Gauss-Like Quadrature

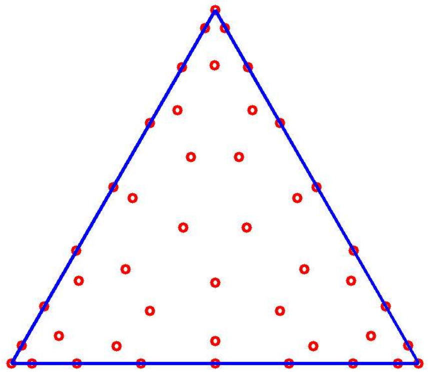
Spectral element methods have integrals of the form:

$$\int_{\Delta} f\,\phi_i \qquad \phi_i \in P^d$$

To evaluate these integrals with quadrature points $\{z_i\}$

$$\int_{\Delta} g = \sum_i w_i\, g(z_i) \qquad \forall\, g \in P^{d+e}$$

Gauss Points: e = d + 1
Gauss-Lobatto Points: e = d - 1
Gauss-Like quadrature: e >> 1



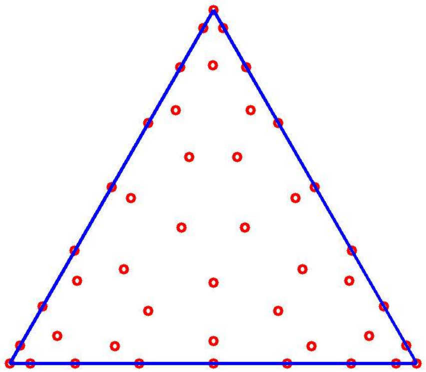
Notation

Coordinate in the triangle:

$$\zeta = (x\, ,\, y) \in \Delta$$

Quadrature points $\{z_i\}$ and weights $\{w_i\}$:

$$z = \left\{ z_{1,} z_{2,} \dots z_N \right\}$$
$$w = \left\{ w_{1,} w_{2,} \dots w_N \right\}$$



Koornwinder Polynomial basis $\{g_j(\zeta)\}$ for P^{d+e}

$$\text{span}\left\{ g_j \right\} = P^{d+e}$$

Traditional Algorithm for Quadrature Points

Let: $$F_k = \sum_i w_i g_k(z_i) - \int_\Delta g_k$$

Solve for quadrature points $\{z_i\}$ and weights $\{w_i\}$:

$$F_k = 0 \qquad \forall k : g_k \in P^{d+e}$$

Symmetry: If the points are invariant under the action of a group (such as D_3), we need only solve $F=0$ for the subspace of P^d invariant under D_3

Wandzura, Xiao, *Symmetric Quadrature Rules on a Triangle,* Comp. Math. App., 2003
Lyness, Jesperson, *Moderate degree symmetric quadrature rules for the triangle*, J. Inst. Math. Appl. 1975

Traditional Algorithm for Quadrature Points

Newton Iteration:

$$\begin{pmatrix} z \\ w \end{pmatrix}_t = -(dF)^{-1} F$$

$$\frac{\partial F_j}{\partial z_i} = w_i \, g'_j(z_i) \qquad \frac{\partial F_j}{\partial w_i} = g_j(z_i)$$

Cardinal Function Algorithm

1. Require $N = \dim P^d$ so we can construct cardinal functions $\{\phi_i\}$

2. Work with spectral representation of cardinal functions:

$$\phi_i(\zeta) = \sum_k \hat{\phi}_i^k g_k(\zeta)$$

3. Define quadrature weights: $w_i = \hat{\phi}_i^0$

Then: $$\int_\Delta g = \sum_i w_i g(z_i) \qquad \forall g \in P^d$$

Cardinal Function Algorithm

Equations:

$$F_k = \sum_i w_i g_k(z_i) = 0 \qquad \forall k : g_k \in P^{d+e} - P^d$$

Newton Iteration: $$z_t = -(dF)^{-1} F$$

Reduced number of unknowns from 3N to 2N
Reduced number of equations by dim P^d = N

Cardinal Function Algorithm

$$\frac{\partial F_j}{\partial z_k} = w_k g'_j(z_k) + \sum_i g_j(z_i) \frac{\partial w_i}{\partial z_k}$$

$$\frac{\partial w_i}{\partial z_k} = ?$$

Cardinal Function Algorithm

$$\frac{\partial \phi_i}{\partial z_k}(\zeta) = -\phi'_i(z_k)\ \phi_k(\zeta)$$

Assuming $w_i = \hat{\phi}_i^0$

$$\frac{\partial w_i}{\partial z_k} = -\phi'_i(z_k)\, w_k$$

Cardinal Function Algorithm

Newton Iteration: $$z_t = -(dF)^{-1} F$$

$$F_k = \sum_i w_i g_k(z_i) \qquad \forall k : g_k \in P^{d+e} - P^d$$

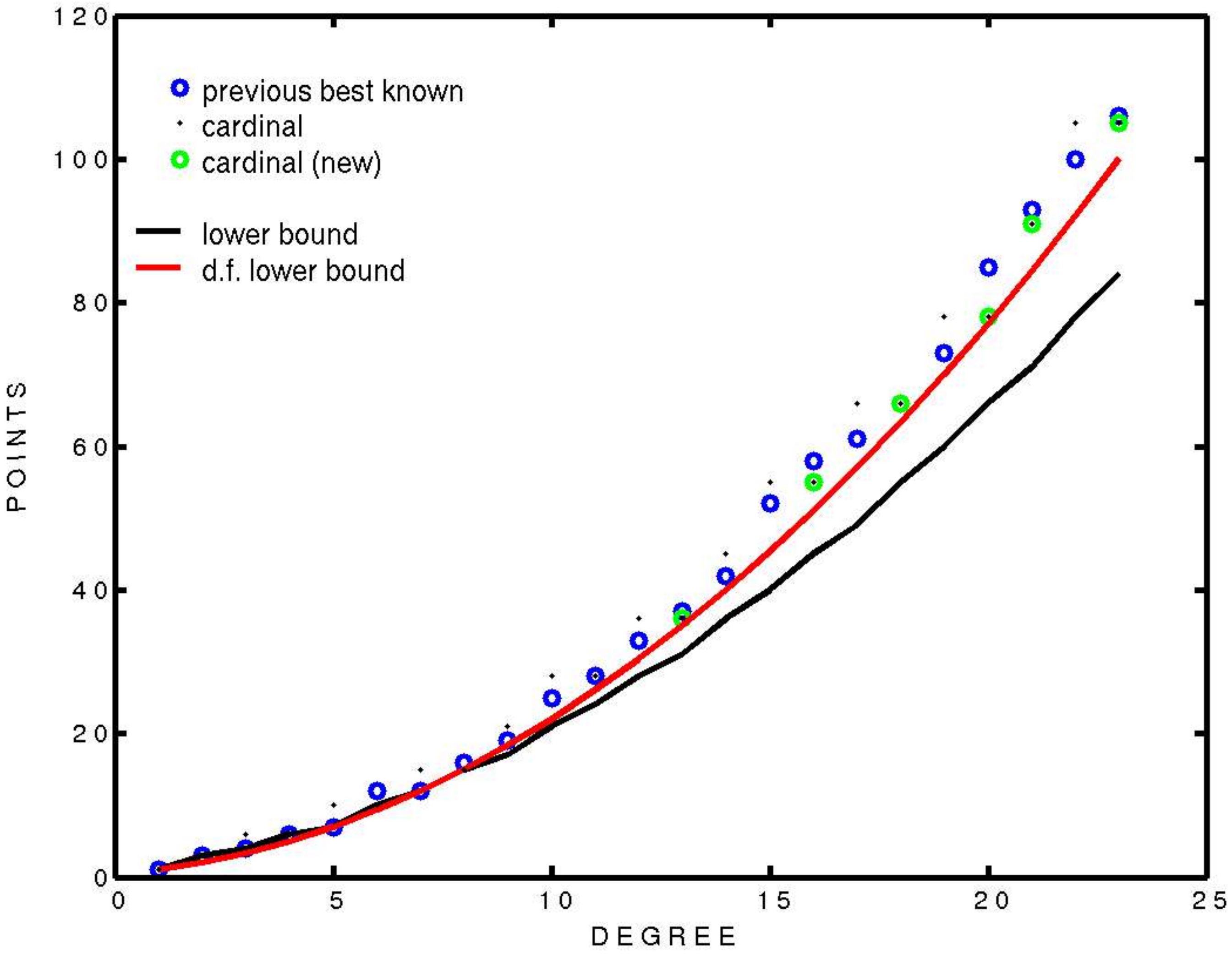
$$w_i = \hat{\phi}_i^0$$

$$\frac{\partial F_j}{\partial z_k} = w_k \left(g'_j(z_k) - \sum_i \phi'_i(z_k) g_j(z_i) \right)$$

Results

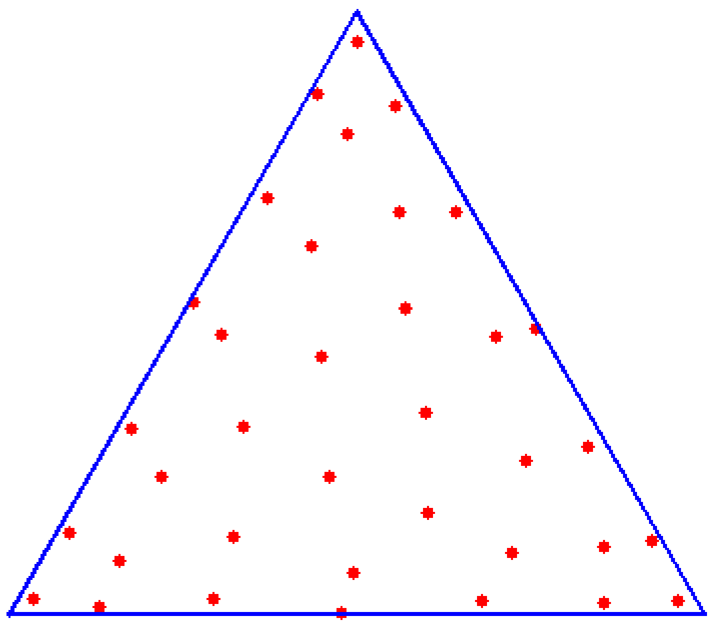
- Solution of cardinal function algorithm gives quadrature formula for P^{d+e} of the form:
  - Cardinal function requirement: $N = \dim P^d$
  - Degrees of freedom requirement: $2N \geq \dim (P^{d+e}\text{-}P^d)$
- Optimal solutions found for all cases tried ($d+e<24$)
- Includes 6 new quadrature sets

Results

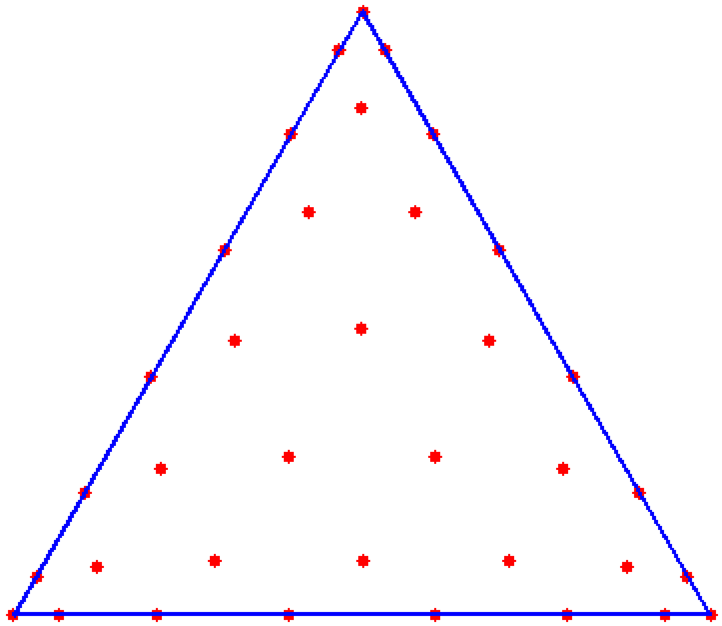


Cools, *An encyclopaedia of cubature formulas,* Journal of Complexity, 2003
Wandzura, Xiao, *Symmetric Quadrature Rules on a Triangle,* Comp. Math. App., 2003

Points for degree 7

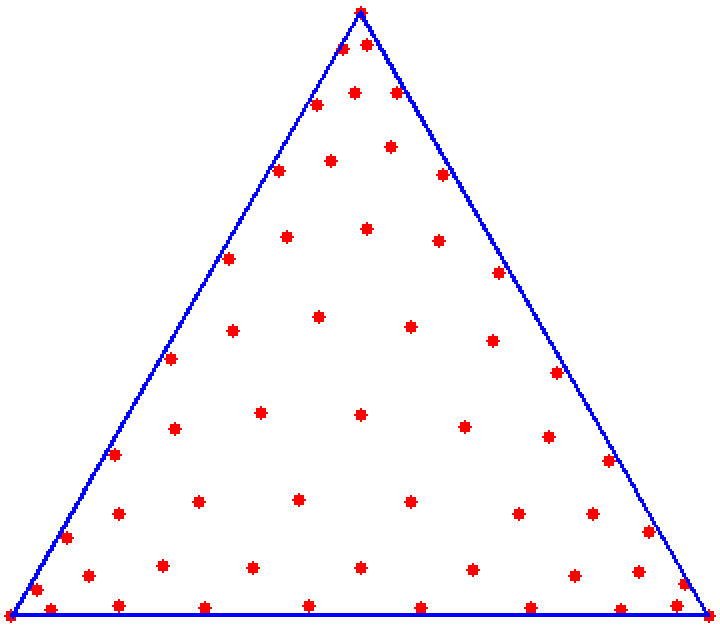


Quadrature (degree 13)

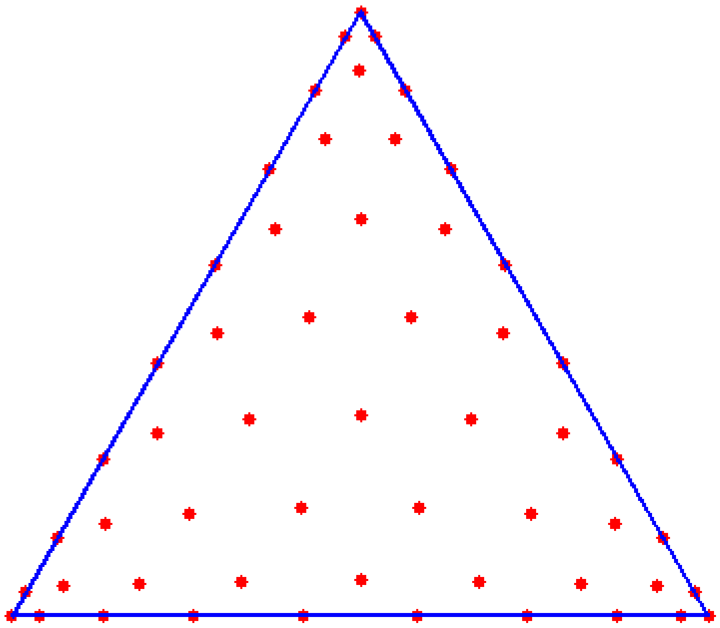


Fekete

Points for degree 9

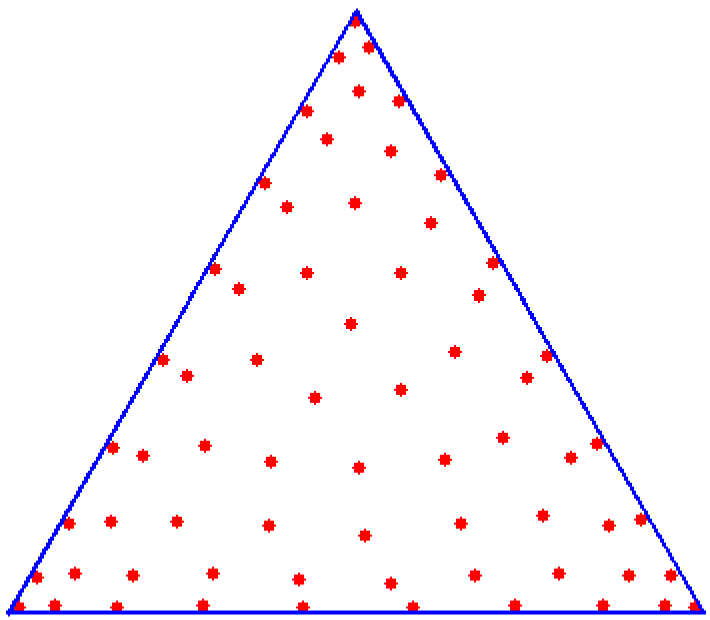


Quadrature (degree 16)

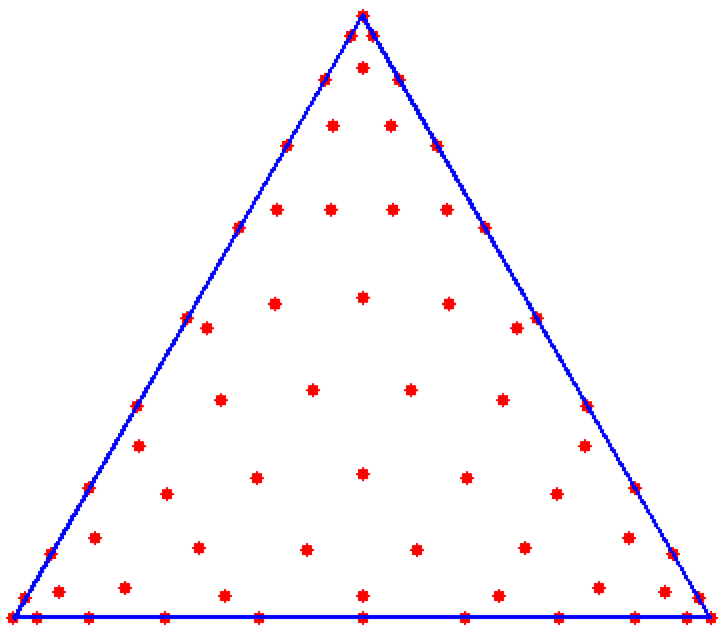


Fekete

Points for degree 10

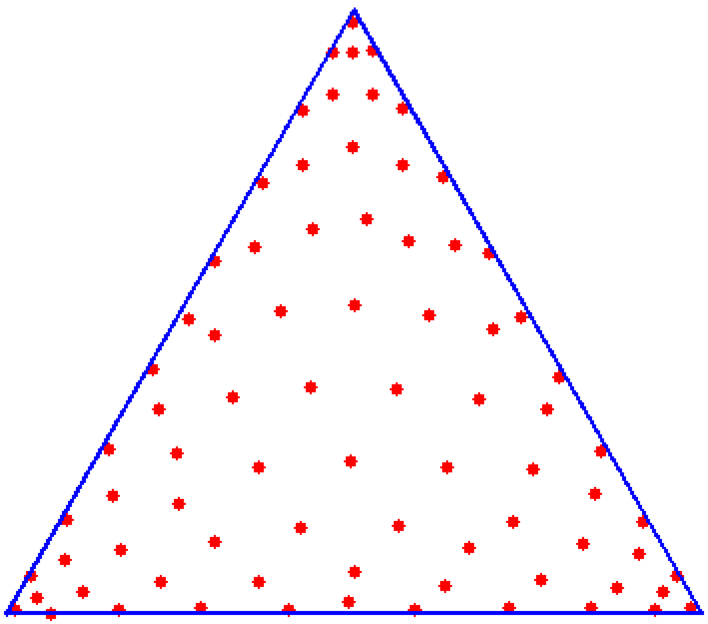


Quadrature (degree 18)

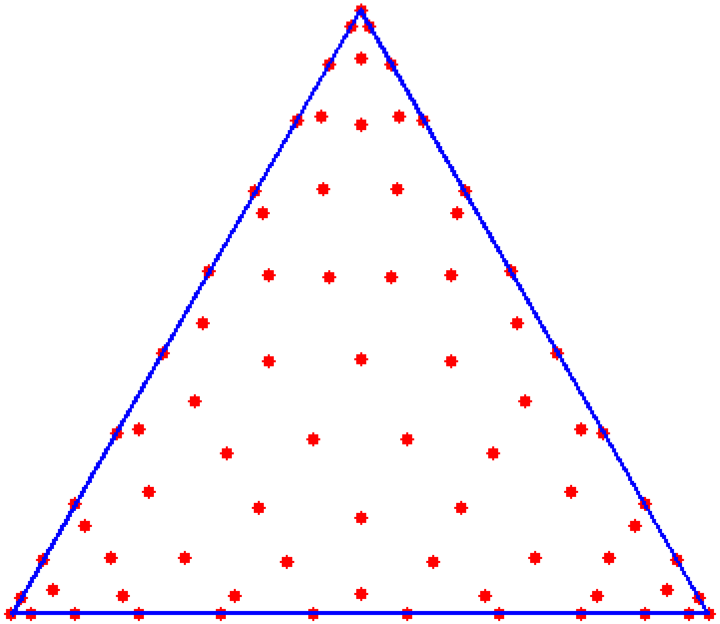


Fekete

Points for degree 11

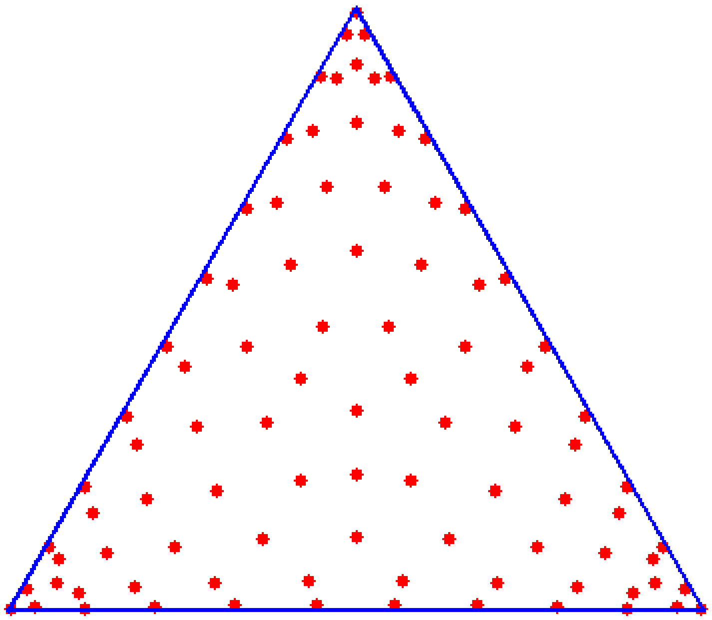


Quadrature (degree 20)

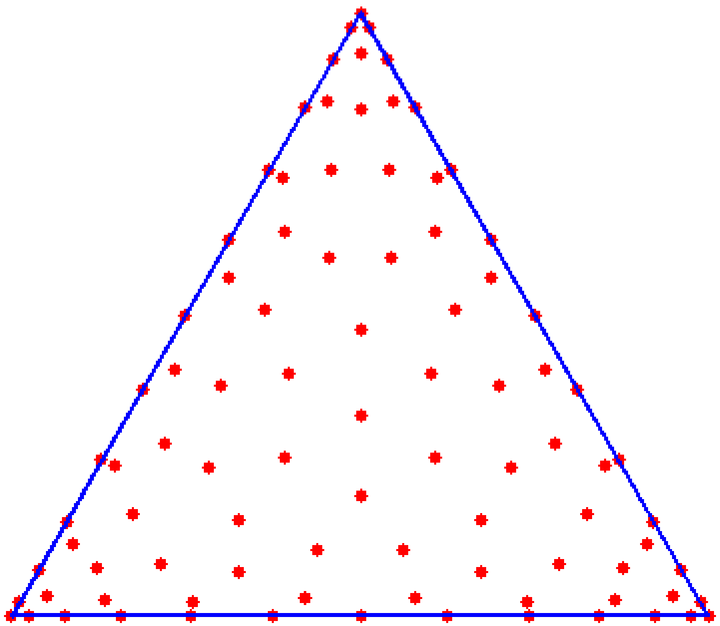


Fekete

Points for degree 12

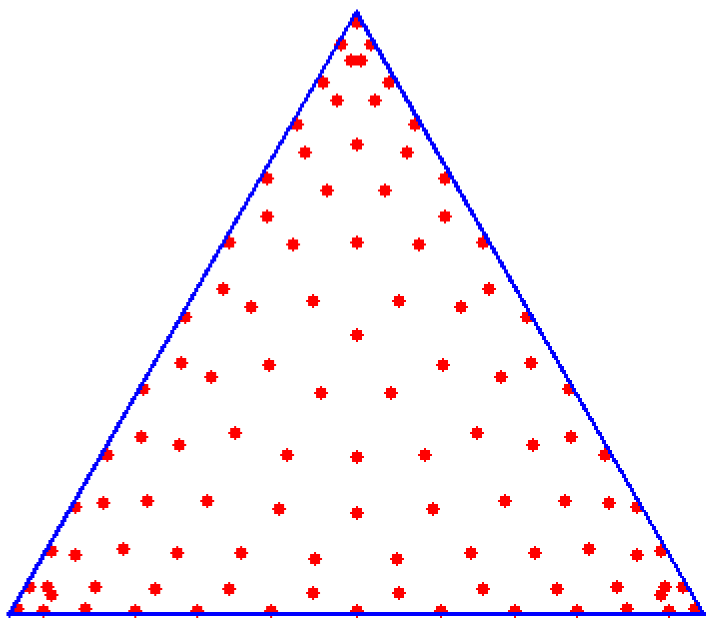


Quadrature (degree 21)

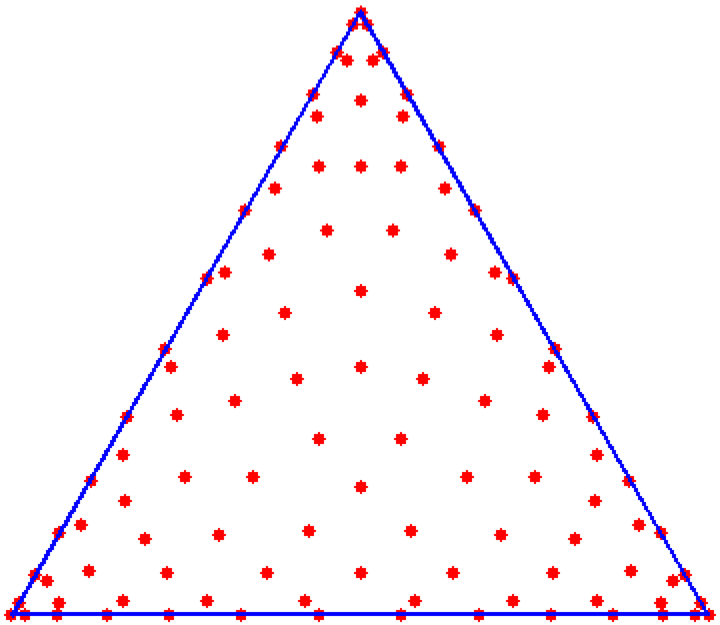


Fekete

Points for degree 13

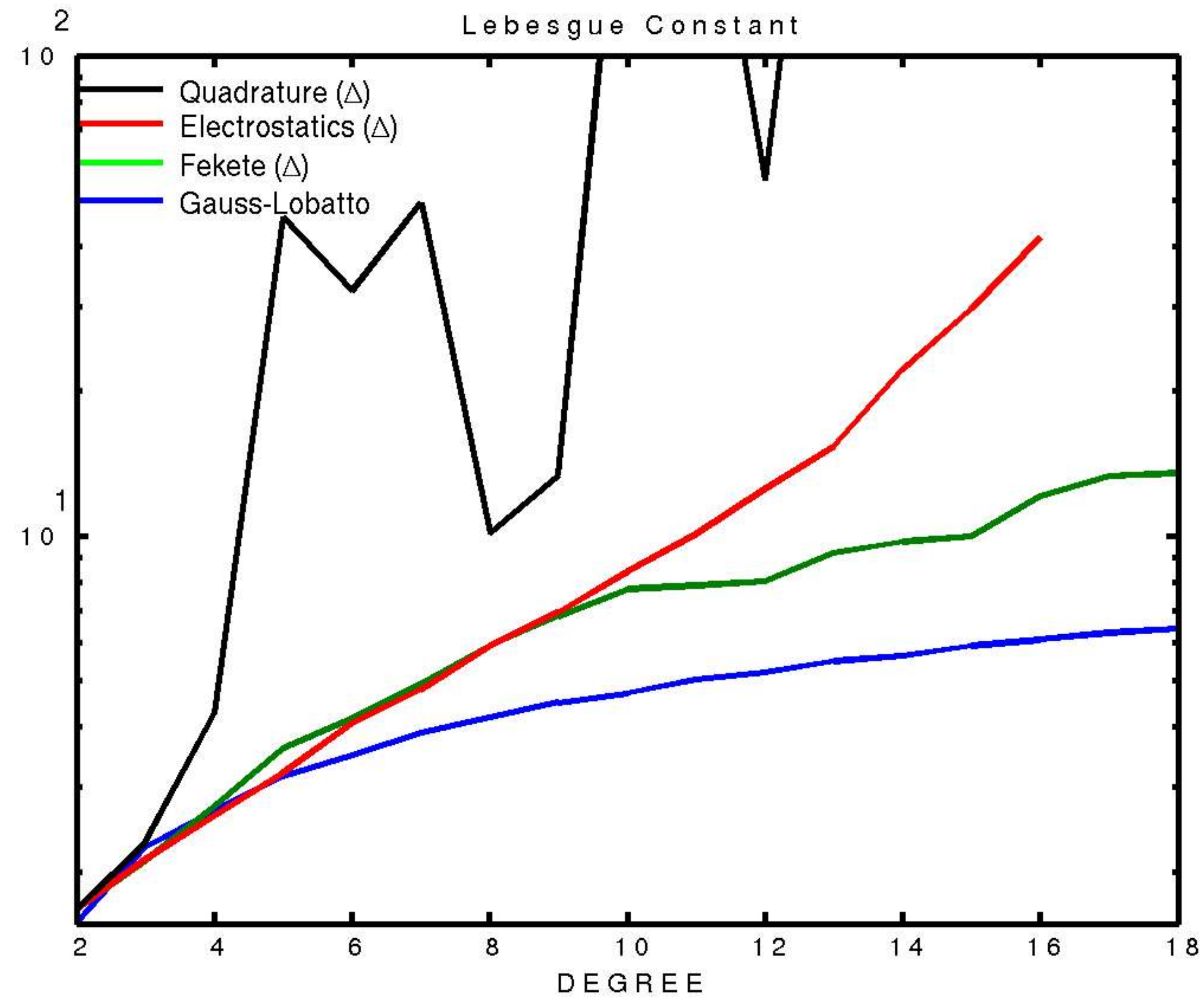


Quadrature (degree 23)



Fekete

Lebesgue Constant



Conclusion

- Cardinal function Quadrature algorithm
  - Optimal formulas found numerically up to degree 23
  - 6 formulas improve on previous best-known results
- Small Lebesgue constant $\neq$ Good Quadrature